**6.15 Power Set by Unique Integers**

**Aim:** The aim is to generate all possible subsets (the power set) from a given set of unique integers and then identify which of those subsets contain a specific target element.

**Algorithm:** 1. Let the input be a set of unique integers: S = {s₁, s₂, …, sn} and a target element t ∈ S.

2. Sort the set S in non-decreasing order.

3. The total number of subsets of S is 2ⁿ.

4. Define a recursive backtracking procedure:  
  GenerateSubsets (start, current, result)  
  - start → the index in S to consider next  
  - current → the subset being built  
  - result → collection of all generated subsets

5. At each call: - Add a copy of current to result.

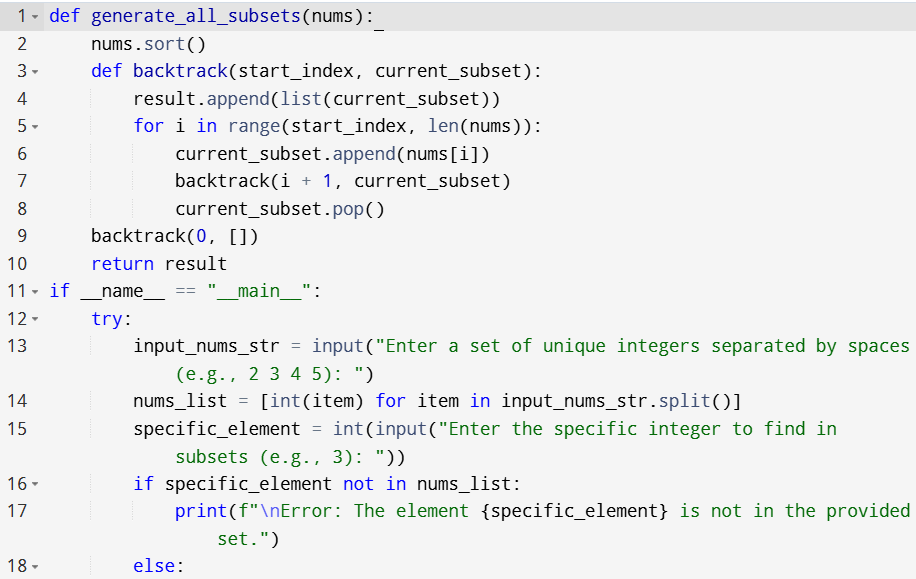
6. Recursive case: - For each index i from start to n−1:  
    a. Include S[i] in current.  
    b. Recursively call:  
      GenerateSubsets (i+1, current, result)  
    c. Backtrack by removing S[i] from current.

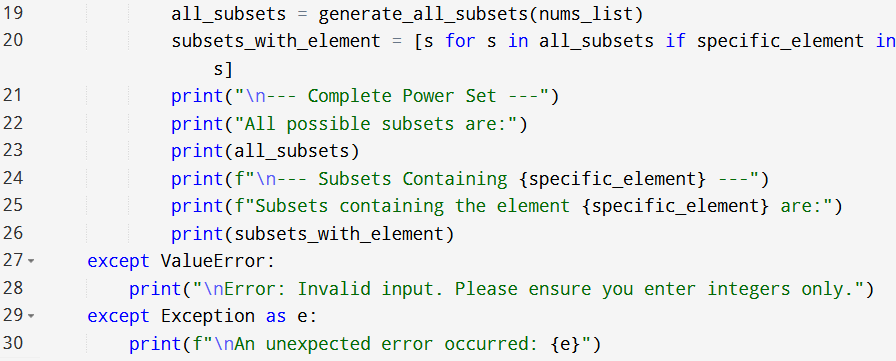
7. Initially, call the procedure as: GenerateSubsets (0, ∅, result)

8. Once the power set result is generated, filter it to select only those subsets that contain the target element t.

9. The final output is the collection of subsets from result that satisfy: t ∈ subset.

**Program:**

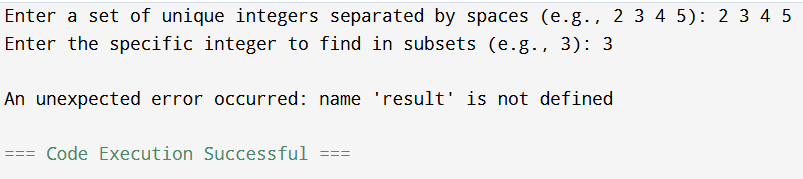
****



**Input:**

* Enter a set of unique integers separated by spaces (e.g., 2 3 4 5):
* Enter the specific integer to find in subsets (e.g., 3):

**Output:**

****

**Result:** Thus, the program is executed successfully and output is verified.

**Performance analysis:**

* Time Complexity: O(n\*2^n)
* Space Complexity: O(n).